IN THE CLAIMS:

Please amend claims 62, 71, 75, 78, 80, 81, and 85, as indicated in the following listing of claims. This listing replaces any previous listing of claims.

Claims 1-27. (Canceled).

28. (Previously presented) A method in a data processing system for accessing network services associated with a lookup service, comprising the steps of:

receiving a request from a client by the lookup service for access to one of the network services, the client being remote with respect to the lookup service; and

returning a resource locator to the client from the lookup service so that the client may dynamically load executable code to facilitate access of the one network service.

29. (Previously presented) The method of claim 28, further comprising the step of:

using the returned resource locator to dynamically load executable code to facilitate access of the one network service.

30. (Previously presented) The method of claim 29, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

31. (Previously presented) The method of claim 28, wherein the step of returning a resource locator includes the step of:

returning stub information to the client.

32. (Previously presented) The method of claim 31, further comprising the step of:

using the resource locator in the client to dynamically load executable code for the stub.

33. (Previously presented) The method of claim 32, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

34. (Previously presented) The method of claim 28, wherein the step of returning a resource locator includes the step of:

returning smart proxy information to the client.

35. (Previously presented) The method of claim 34, further comprising:

using the resource locator in the client to dynamically load executable code for a smart proxy.

36. (Previously presented) The method of claim 35, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

37. (Previously presented) A method in a data processing system having a lookup service with a plurality of services, the method comprising the steps of:

sending from a client to the lookup service a request to access one of the services, the lookup service being remote with respect to the client; and

receiving, responsive to the request, by the client from the lookup service, a resource locator to dynamically load executable code to facilitate access of the one service.

38. (Previously presented) The method of claim 37, further comprising the step of:

using the received resource locator to dynamically load executable code to facilitate access of the one network service.

39. (Previously presented) The method of claim 38, further comprising the step of:

accessing the service by the client using the dynamically loaded executable code.

40. (Previously presented) The method of claim 37, wherein the step of receiving a resource locator comprises the step of:

receiving stub information; and

using the resource locator to dynamically load executable code for the stub.

41. (Previously presented) The method of claim 40, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

42. (Previously presented) The method of claim 37, wherein the step of receiving a resource locator comprises the step of:

receiving smart proxy information; and

using the resource locator to dynamically load executable code for a smart proxy.

43. (Previously presented) The method of claim 40, further comprising the step of:

accessing the network service by the client using the dynamically loaded executable code.

44. (Previously presented) A distributed system with a plurality of network services, comprising:

a server computer with a lookup service having a resource locator corresponding to dynamically executable code for facilitating access to one of the network services; and

a client computer with a program that sends a request to the lookup service for the one network service, that receives the resource locator from the server computer, and uses the resource locator to dynamically load executable code to facilitate access of the one network service.

45. (Previously presented) A data processing system for accessing network services associated with a lookup service, comprising:

means for receiving a request from a client by the lookup service for access to one of the network services, the client being remote with respect to the lookup service; and

means for returning a resource locator to the client from the lookup service such that the client may dynamically load executable code to facilitate access of the one network service.

46. (Previously presented) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing network services associated with a lookup service, the method comprising the steps of:

receiving a request from a client by the lookup service for access to one of the network services, the client being remote with respect to the lookup service; and

returning a resource locator to the client from the lookup service so that the client may dynamically load executable code to facilitate access of the one network service.

47. (Previously presented) The computer-readable medium of claim 46, wherein the method further comprises the step of:

using the returned resource locator to dynamically load executable code to facilitate access of the one network service.

48. (Previously presented) The computer-readable medium of claim 47, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

49. (Previously presented) The computer-readable medium of claim 46, wherein the step of a returning a resource locator comprises the step of:

returning stub information to the client.

50. (Previously presented) The computer-readable medium of claim 49, wherein the method further comprises the step of:

using the resource locator in the client to dynamically load executable code for the stub.

51. (Previously presented) The computer-readable medium of claim 50, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

52. (Previously presented) The computer-readable medium of claim 46, wherein the step of returning a resource locator comprises the step of:

returning smart proxy information to the client.

53. (Previously presented) The computer-readable medium of claim 46, wherein the step of returning a resource locator comprises the step of:

using the resource locator in the client to dynamically load executable code for a smart proxy.

54. (Previously presented) The computer-readable medium of claim 53, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

55. (Previously presented) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing network services associated with a lookup service, the method comprising the steps of:

sending from a client to the lookup service a request to access one of the services, the lookup service being remote with respect to the client; and

receiving, responsive to the request, by the client from the lookup service, a resource locator to dynamically load executable code to facilitate access of the one service.

56. (Previously presented) The computer-readable medium of claim 55, wherein the method further comprises the step of:

using the received resource locator to dynamically load executable code to facilitate access of the one network service.

57. (Previously presented) The computer-readable medium of claim 56, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

58. (Previously presented) The computer-readable medium of claim 55, wherein the step of receiving a resource locator includes the step of:

receiving stub information; and

using the resource locator to dynamically load executable code for the stub.

59. (Previously presented) The computer-readable medium of claim 58, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

60. (Previously presented) The computer-readable medium of claim 55, wherein the step of receiving a resource locator includes the step of:

receiving smart proxy information; and

using the resource locator in the client to dynamically load executable code for a smart proxy.

61. (Previously presented) The computer-readable medium of claim 60, wherein the method further comprises the step of:

accessing the network service by the client using the dynamically loaded executable code.

62. (Currently Amended) A method in a data processing system for accessing network services associated with a lookup service, comprising the steps of:

receiving a request by the lookup service for access to one of the network services; and

returning, by the lookup service, executable code to facilitate the access to the one network service, wherein the executable code is configured to perform at least one of storing data obtained from the one network service for subsequent use and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission.

63. (Previously Presented) The method of claim 62, further including the step of:

accessing the network service using the returned executable code.

64. (Previously Presented) The method of claim 62, wherein the step of returning the executable code includes the step of:

returning a stub embedded within the executable code.

65. (Previously Presented) The method of claim 62, wherein the executable code returns the stored data to a system requesting the data from the one network service following the generating step.

66. (Previously Presented) The method of claim 62, wherein the receiving step includes the step of:

receiving the request to access the one network service such that the one network service is identified by a type of service.

67. (Previously Presented) The method of claim 62, wherein the receiving step includes the step of:

receiving the request to access the one network service such that the one network service is identified by a service identification number.

68. (Previously Presented) The method of claim 62, wherein the receiving step includes the step of:

receiving the request to access the one network service such that the one network service is identified by an attribute of the one network service.

69. (Previously Presented) The method of claim 62, wherein the step of returning executable code includes the step of:

returning a plurality of stubs included in respective smart proxies associated with a plurality of network services that match the request.

70. (Previously Presented) The method of claim 69, wherein the step of returning a plurality of stubs includes the step of:

returning attributes associated with each stub.

71. (Currently Amended) A method in a data processing system having a lookup service with a plurality of services, the method comprising the steps of:

sending to the lookup service a request to access one of the services; and

receiving executable code, from the lookup service and responsive to the request, that facilitates use of the one service wherein the executable code is_ configured to perform at least one of storing data obtained from the one service for subsequent use and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission.

72. (Previously Presented) The method of claim 71, further including the step of:

accessing the one service using the received executable code.

73. (Previously Presented) The method of claim 71, wherein the step of receiving executable code includes the step of:

receiving a stub that facilitates access of the one service.

74. (Previously Presented) The method of claim 73, wherein the executable code is a smart proxy including the stub.

75. (Currently Amended) A method in a data processing system having a first computer with a client and a second computer with a lookup service containing service stubs used for accessing associated services, the method comprising the steps of:

sending a request by the client to the lookup service identifying one of the associated services to be accessed;

receiving the request by the lookup service;

searching the lookup service for the identified service;

returning a service stub associated with the identified service to the client, wherein the service stub is embedded in executable code configured to perform at least one of storing data obtained from the one network service for subsequent access and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission;

receiving the service stub by the client;

loading the service stub into an address space of the client to render the service stub available for use to invoke the identified service; and

using the stub by the client to access the identified service.

76. (Previously Presented) The method of claim 75, wherein the sending step includes the step of:

sending a request for more than one service to be returned, and wherein the returning step further includes the step of:

returning more than one stub to the client in response to the request.

77. (Previously Presented) The method of claim 75, wherein the searching step includes the step of:

returning a null value if no service is found matching the request.

78. (Currently Amended) A distributed system with a plurality of network services, comprising:

a server computer with a lookup service having stubs for facilitating access to the network services; and

a client computer with a program that sends a request to the lookup service for one of the stubs corresponding to one of the network services, that receives the stub, and that accesses the one network service using the received stub, wherein the stub is embedded in executable code configured to perform at least one of storing data obtained from the one network service for subsequent access and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission.

79. (Previously Presented) The distributed system of claim 78, wherein the client computer utilizes a remote procedure call mechanism to receive the stub.

80. (Currently Amended) A system having a first computer with a client and a second computer with a lookup service containing service stubs used for accessing associated services, the system comprising:

means for sending a request by the client to the lookup service identifying one of the associated services to be accessed;

means for receiving the request by the lookup service;

means for searching the lookup service for the identified service;

means for returning a service stub associated with the identified service to the client, wherein the service stub is embedded in executable code configured to perform at least one of storing data obtained from the one network service for subsequent access and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission;

means for receiving the service stub by the client;

means for loading the service stub into an address space of the client to render the service stub available for use to invoke the identified service; and

means for using the stub by the client to access the identified service.

81. (Currently Amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing network services contained in a lookup service, the method comprising the steps of:

receiving a request by the lookup service for access to one of the network services; and returning, by the lookup service, executable code to facilitate the access to the one network service, wherein the executable code is configured to perform at least one of storing data obtained from the one network service for subsequent use and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission.

82. (Previously Presented) The computer-readable medium of claim 81, wherein the method further includes the step of:

accessing the network service using the returned executable code.

83. (Previously Presented) The computer-readable medium of claim 81, wherein the returning step includes the step of:

returning a stub embedded in the executable code to the client.

84. (Previously Presented) The computer-readable medium of claim 81, wherein the executable code returns the stored data to a system requesting the data from the one network service following the generating step.

85. (Currently Amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method for accessing network services contained in a lookup service, the method comprising the steps of:

sending to the lookup service a request to access one of the services; and receiving executable code, from the lookup service and responsive to the request, that facilitates use of the one service, wherein the executable code is configured to perform at least one of storing data obtained from the one service for subsequent use and transforming parameters associated with the request into different types of parameters appropriate for subsequent transmission.

86. (Previously Presented) The computer-readable medium of claim 85, wherein the method further includes the step of:

accessing the one service using the received executable code.

87. (Previously Presented) The computer-readable medium of claim 85, wherein the step of receiving executable code includes the step of:

receiving a stub that facilitates access of the one service.

88. (Previously Presented) The computer-readable medium of claim 85, wherein the executable code is a smart proxy including the stub.